In [1]:

```
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
```

```
<frozen importlib._bootstrap>:228: RuntimeWarning: scipy._lib.messagestre
am.MessageStream size changed, may indicate binary incompatibility. Expec
ted 56 from C header, got 64 from PyObject
```

In [2]:

```
df = pd.read_csv("Downloads/Automobile_data.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wh b
0	3	?	alfa-romero	gas	std	two	convertible	rwd	front	
1	3	?	alfa-romero	gas	std	two	convertible	rwd	front	
2	1	?	alfa-romero	gas	std	two	hatchback	rwd	front	
3	2	164	audi	gas	std	four	sedan	fwd	front	
4	2	164	audi	gas	std	four	sedan	4wd	front	

5 rows × 26 columns

In [4]:

```
df.columns
```

Out[4]:

```
Index(['symboling', 'normalized-losses', 'make', 'fuel-type', 'aspiratio
n',
       'num-of-doors', 'body-style', 'drive-wheels', 'engine-location',
       'wheel-base', 'length', 'width', 'height', 'curb-weight', 'engine-
type',
       'num-of-cylinders', 'engine-size', 'fuel-system', 'bore', 'strok
e',
       'compression-ratio', 'horsepower', 'peak-rpm', 'city-mpg',
       'highway-mpg', 'price'],
      dtype='object')
```

In [5]:

```
df.shape
```

Out[5]:

```
(205, 26)
```

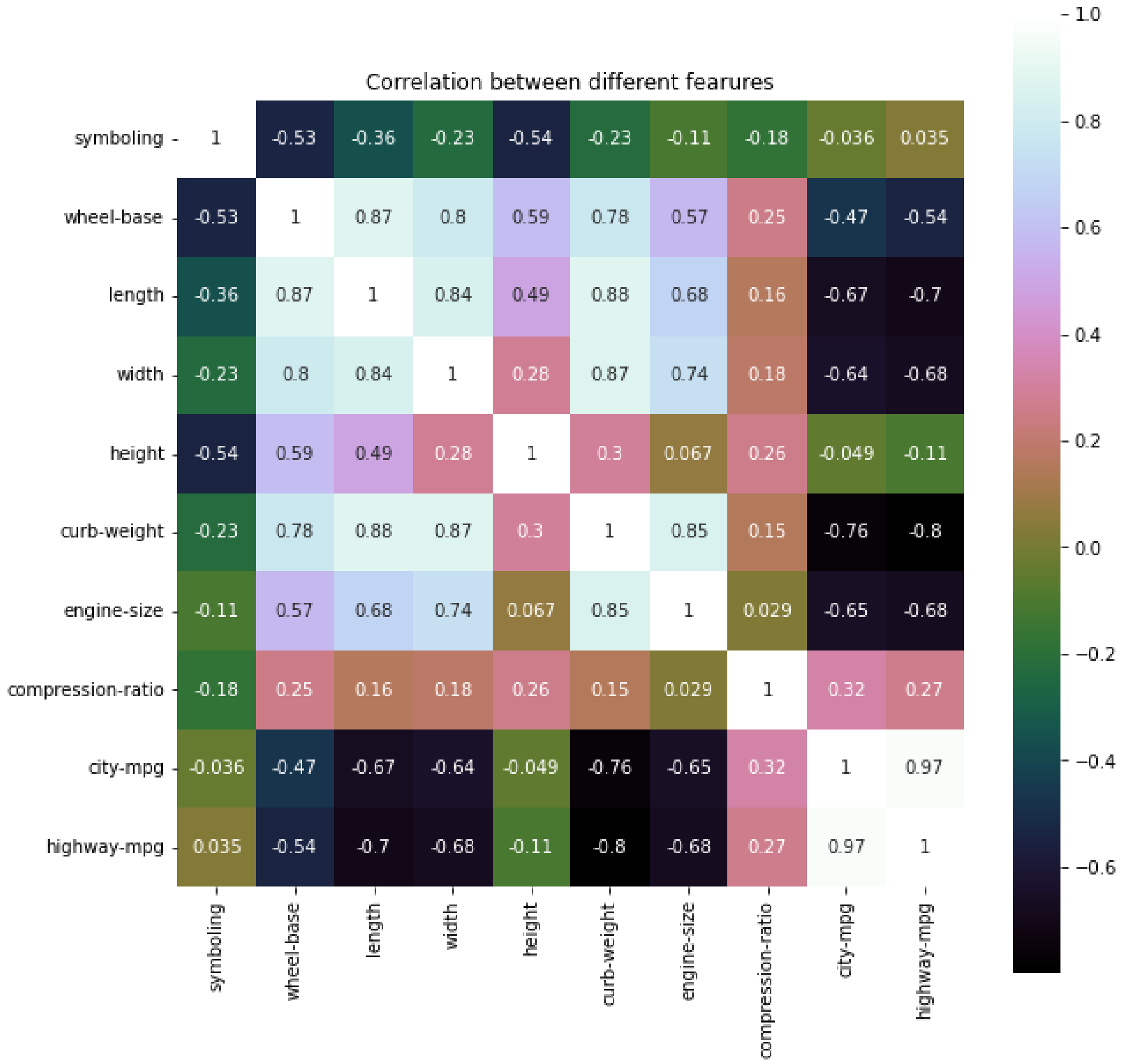
In [6]:

```
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
correlation = df.corr()
plt.figure(figsize=(10,10))
sns.heatmap(correlation, vmax=1, square=True,annot=True,cmap='cubehelix')

plt.title('Correlation between different fearures')
```

Out[6]:

```
Text(0.5, 1.0, 'Correlation between different fearures')
```



In [7]:

```
set(df._get_numeric_data().columns)
```

Out[7]:

```
{'city-mpg',
 'compression-ratio',
 'curb-weight',
 'engine-size',
 'height',
 'highway-mpg',
 'length',
 'symboling',
 'wheel-base',
 'width'}
```

In [8]:

```
#checking which columns are of categorical data type
categorical_features=set(df.columns)-set(df._get_numeric_data().columns)
categorical_features
```

Out[8]:

```
{'aspiration',
 'body-style',
 'bore',
 'drive-wheels',
 'engine-location',
 'engine-type',
 'fuel-system',
 'fuel-type',
 'horsepower',
 'make',
 'normalized-losses',
 'num-of-cylinders',
 'num-of-doors',
 'peak-rpm',
 'price',
 'stroke'}
```

In [9]:

```
df = df.replace('?', 0)
```

In [10]:

```
X = df.drop("price", axis = 1)
X = pd.get_dummies(X , drop_first = True)
X = X.replace("?", 0)
```

In [11]:

```
from sklearn.preprocessing import StandardScaler
X = StandardScaler().fit_transform(X)
pca = PCA(n_components=10)
X_pca = pca.fit_transform(X)
```

In [12]:

```
X_pca
```

Out[12]:

```
array([[-0.25503705,  3.24958888,  0.31795578, ..., -1.32172256,
        -0.54310855,  0.69970158],
       [-0.25503705,  3.24958888,  0.31795578, ..., -1.32172256,
        -0.54310855,  0.69970158],
       [ 2.03578618,  4.06272205,  0.48441795, ..., -0.04989506,
        -1.33449003,  0.12268619],
       ...,
       [ 6.35367925,  0.49196164, -0.57254854, ...,  0.04406385,
        -0.82153481, -3.44050848],
       [ 4.83795103, -4.28189249,  0.45451469, ...,  1.59476195,
         0.63364865, -0.81042478],
       [ 4.28188581, -1.6584463 , -2.43404809, ...,  1.14408753,
        -1.30063567, -5.12519929]])
```

In [13]:

```
print(pca.components_)
```

```
[[-0.08321815  0.22863867  0.24662451 ...  0.02572243 -0.04235762
  -0.0059824 ]
 [ 0.18205109 -0.11915763 -0.04192003 ...  0.15131588  0.0585352
   0.02733746]
 [ 0.15878273 -0.02163608 -0.02776901 ... -0.02254581  0.09425841
  -0.01146713]
 ...
 [-0.00624242 -0.01543869 -0.01040589 ...  0.18331766 -0.11109015
   0.01292066]
 [ 0.08812847 -0.01625966  0.03404654 ...  0.05925213  0.07065234
  -0.09173645]
 [ 0.11687466 -0.00096345  0.02380411 ... -0.08839002 -0.06576888
   0.12887002]]
```

In [16]:

```
exp_var = pca.explained_variance_
print(exp_var)
```

```
[13.113643    8.87550014  6.84188679  6.73202057  6.38977514  6.01155834
  5.78937817  5.7095815   5.54146309  5.35548975]
```

In [15]:

```
pca.fit(X)
X_pca = pca.transform(X)
print("original shape:   ", X.shape)
print("transformed shape:", X_pca.shape)
```

```
original shape:    (205, 268)
transformed shape: (205, 10)
```

In [17]:

```
total_variance = sum(exp_var)
variance_ratio = [(i/total_variance)*100 for i in exp_var]
data = pd.DataFrame({'Variance Ratio(%)': variance_ratio})
data.index += 1
print(data)
```

```
    Variance Ratio(%)
1           18.637845
2           12.614359
3            9.724073
4            9.567925
5            9.081507
6            8.543964
7            8.228189
8            8.114777
9            7.875838
10           7.611522
```

Conclusion: Hence, We can say that highest variance is explained by PCA 1 with variance Ratio Percentage of 18.637845 % out of the 10 PCA Components.